

**REMARKS**

Reconsideration of the above-identified application is respectfully requested.

Claims 1-54 are pending in the present application.

In the Office Action of April 2, 2004, which has been made FINAL, the Examiner rejected Claims 1-2 and 19-20 under 35 U.S.C. §102(e), as allegedly being anticipated by Krishnaswamy et al. (U.S. Patent No. 6,622,300)(hereinafter "Krishnaswamy"). The Examiner additionally rejected Claims 3-8, 11, 18, 21-27, 30, 36-45, 48 and 54 under 35 U.S.C. §103(a), as allegedly being unpatentable over Krishnaswamy in view of Holzle et al. (U.S. Patent No. 5,995,754) (hereinafter "Holzle1"). The Examiner additionally rejected Claims 9, 28 and 46 under 35 U.S.C. §103(a), as allegedly being unpatentable over Krishnaswamy in view of Holzle1 and further in view of O'Donnell (U.S. Patent No. 6,374,369) (hereinafter "O'Donnell"). The Examiner additionally rejected Claims 10, 29 and 47 under 35 U.S.C. §103(a), as allegedly being unpatentable over Krishnaswamy in view of Holzle1 and further in view of Benitez (U.S. Patent No. 6,189,141) (hereinafter "Benitez"). The Examiner additionally rejected Claims 12, 31 and 49 under 35 U.S.C. §103(a), as allegedly being unpatentable over Krishnaswamy in view of Holzle1 and further in view of Ronstrom (U.S. Patent Publication No.2002/0010913) (hereinafter "Ronstrom"). The Examiner additionally rejected Claim 13 under 35 U.S.C. §103(a), as allegedly being unpatentable over Krishnaswamy in view of the reference to Alpern et al. entitled "The Jalapeno Virtual Machine", IBM Systems Journal, Vol. 39, No. 1, February 2000 (hereinafter "Alpern"). Finally, the Examiner rejected Claims 14-17, 32-35, and 50-53 under 35 U.S.C. §103(a), as allegedly being unpatentable over Krishnaswamy in view of the reference to Urs Hölzle, et al entitled "Reconciling Responsiveness with Performance in Pure Object-Oriented Languages", by, ACM Transactions on Programming Languages and Systems, Vol. 18, No. 4, July 1996, pp. 355-400 (hereinafter "Holzle2").

With respect to the substantive rejections of independent Claims 1 and 19 under 35 U.S.C. §102(e), the Applicants' respectfully amend each of independent Claims 1, 19 and 37 to set forth claim limitations that clarify the inventive features previously argued in applicants' arguments submitted in their prior response of January 28, 2004. Particularly, Claims 1, 19 and 37 are being amended to set forth that a system, method and computer program product for adaptively optimizing a computer program executing in a virtual machine execution environment, the virtual machine execution environment comprising one or more compiler devices for providing various levels of program optimization. The method and computer program executes steps comprising: a) sampling the executing computer program to obtain raw profile data samples; b) characterizing the raw profile data as meeting a threshold criteria; c) analyzing the characterized raw profile data for determining whether a predetermined level of program optimization for the executing program is to be performed by a compiler device, and generating a compilation plan in accordance with a determined level of optimization; and, d) when optimization is to be performed, invoking a compiler device for optimizing the executing program in accordance with the compilation plan.

Thus, the thrust of the applicants' traversal is that Krishnaswamy wholly teaches away from the adaptive optimizing approach for optimizing performance of a computer program as it does not teach or suggest executing in a virtual machine execution environment, as in the present invention. Krishnaswamy's teaching and claims focus on a dynamic optimization system that uses a kernel module (See Krishnaswamy Claim 1 at Col. 11, lines 11 and 20) and kernel memory space (Krishnaswamy Claim 1 at Col. 11, line 28).

The present invention, on the other hand, is directed to dynamic optimization that is not limited to kernel space. For example, as stated at page 5, lines 25, et seq. of the present patent application, "It is an object of the present invention to provide an adaptive optimization

system for a Java Virtual Machine that implements a sampling technique having lower overhead than invocation counters and that drives adaptive and online feedback-directed optimizations.” Respectfully, no new matter is being entered by incorporation of this amendment to claim 1 as full support is provided in the cited passage in the specification. Furthermore, virtual machines are non-kernel programs, and, although the present specification explicitly mentioned Java virtual machines by way of example, it is applicable to other virtual machines, and thus, the claim is not limited to Java virtual machines.

As Krishnaswamy's patent only applies to kernel systems, it is respectfully submitted that Krishnaswamy does not anticipate the amended Claims 1, 19 and 37.

The Examiner further alleges that Krishnaswamy also teaches the second element of Claim 1, i.e., the element directed to “a controller device for receiving said characterized raw profile data from said runtime measurements sub-system and analyzing said data for determining whether a level of program optimization for said executing program is to be performed by a compiler device, said controller generating a compilation plan in accordance with a determined level of optimization. Namely, the Examiner summarily concludes in regard to Krishnaswamy that *“since the optimization is performed dynamically using the profile data, various levels of optimization is inherently performed depending on the result of the analysis of the profile data”*.

Applicants respectfully do not agree with this assessment in view of the amendment to Claims 1 (19 and 37), which are directed to analyzing the characterized raw profile data for determining whether a predetermined level of program optimization for the executing program is to be performed. That is, the level of optimization set forth in the claims is directed to general optimization levels of a compiler, which are a collection of optimizations that are not directly dependent on profile data, but characterizations thereof (See page 21, line 27 - page 22 line 15, of the present patent application). Thus, this second element of Claim 1

describes choosing which of these levels to perform. The present specification teaches (at page 21, line 27 – page 22 line 15, for example) how a compiler may group optimizations into various optimization levels that are predetermined when the compiler is written. Respectfully, while the Examiner has demonstrated that the initial claims may be interpreted in a more general, unintended manner than as taught in the specification, Claims 1, 19 and 37 are being updated to be consistent with the specification. Respectfully, no new matter is being entered by incorporation of this amendment to Claim 1 as full support is provided in the cited passage in the specification.

As Krishnaswamy does not cover this level of optimization, it is respectfully submitted that Krishnaswamy does not anticipate the amended Claims 1, 19 and 37.

With respect to the rejection of independent Claim 19 as being anticipated by Krishnaswamy, applicants respectfully disagree as Krishnaswamy does not teach a sampling technique as implemented in the present invention but rather, at col. 6, lines 21-34, suggests a low-level “sampling” mechanism such as collecting data from the processor’s Performance Monitoring Units) PMU’s. Further, the arguments submitted hereinabove with respect to the traversal of the rejected claimed elements of Claim 1 are applicable in traversal of the rejection of Claim 19.

Thus, as independent Claims 1 and 19 set forth novel features of a complete operative system and method for adaptively optimizing a computer program executing in an execution environment, that operates in user space (not kernel space) and thus, having elements that are not taught by Krishnaswamy, the Examiner is respectfully requested to withdraw the rejection of Claims 1 and 19 under 35 U.S.C. §102(e) and all claims dependent therefrom.

Furthermore, with respect to the Examiner's rejection of Claims 3-8, 11, 18, 21-27, 30, 36-45, 48, and 54 as being obvious in light of the teachings of Krishnaswamy and Holze1, applicants respectfully disagree.

Respectfully, although the system of Krishnaswamy implements a low-level sampling mechanism to collect data from the processor PMU's, the task of mapping from such low-level profile data (traces of executed binary instructions) to the kind of information required for Holze1's techniques (which are directed to method activations; method invocations, call edges) is a difficult task, even for those quite skilled in the art because the necessary translation information (between high and low level) is not typically preserved because of space efficiency reasons. In the present FINAL REJECTION, the Examiner did not address this argument.

Furthermore, Holze1 does not teach sampling. Rather, in the Holze1 system, a counter is incremented unconditionally every time particular points in the program execution are reached. In the sampling disclosed in the present invention, a counter is incremented only some of the times each particular point in the program execution is reached. This is an important distinction because a key innovation of the invention as set forth in Claims 1 and 19 is the effective use of profile data sampling to guide dynamic optimization in a virtual machine. The techniques that must be used by a dynamic optimization system to effectively use sampled values instead of exhaustively counted values are substantially different, and thus the teaching of Holze1 directed to exploiting exhaustively counted values could not be applied in the context of Krishnaswamy's system, even by one skilled in the art, without undue experimentation.

In the FINAL REJECTION, the Examiner further stated that Holze1's technique is applicable to conditionally incrementing a counter. However, such an extension is not obvious to one skilled in the art because the condition needs to be specified that is being tested against.

Holzle1 does not specify the condition and there is no obvious condition to be used. Thus, Holzle1 does not present a technique for conditionally capturing samples. As further evidence of unobviousness in view of the combination of Krishnaswamy over Holzle1, applicants' respectfully submit that the point of implementing a sampling mechanism over using exhaustive profiling is to reduce the overhead incurred. It is not clear how simply changing an unconditional increment to a conditional increment will not increase overhead.

Therefore the combination of the teachings of Krishnaswamy and Holzle1 would require undue experimentation and would not in fact be obvious to combine even to one skilled in the art.

For this reason, the Examiner is respectfully requested to withdraw the rejection of Claims 3-8, 11, 18, 21-27, 30, 36-45, 48, and 54 as being obvious in light of the teachings of Krishnaswamy and Holzle1.

The foregoing remarks in traversal of the rejection of Claims 3-8, 11, 18, 21-27, 30, 36-45, 48, and 54 as being obvious in light of the teachings of Krishnaswamy and Holzle1 are further applicable to the FINAL REJECTION of Claims 14-17, 32-35, 50-53 based on obviousness over Krishnaswamy in view of Holzle2.

Furthermore, with specific regard respect to the Examiner's FINAL REJECTION of Claims 9, 28 and 46 as being obvious over Krishnaswamy in view of Holzle1 and O'Donnell, Applicants respectfully disagree. In applicants' previous response, applicants demonstrated that the Examiner's cited quote in O'Donnell discusses how a software developer would optimize their software, whereas, it was argued that the claims of the present invention discuss an automatic process with no human intervention for optimizing software. The Examiner's reply

states that *"There is no mention in the claim that "inserting intrusive profiling" must be "automatically performed (no human intervention)"*.

Applicants' respectfully disagree and submit that Claim 9 is derived from the modified Claim 1 that now sets forth: "A sampling-based system for adaptively optimizing a computer program executing in a virtual machine execution environment". It is respectfully submitted that a virtual machine execution environment is an automatic process; a program runs as specified, with no human intervention. Thus, O'Donnell is not applicable.

As such, the Examiner is respectfully requested to withdraw the rejections of Claims 9, 28 and 46.

Furthermore, with specific regard to the rejection of Claim 16, applicants' had previously argued in their prior response that Claim 16 of the invention sets forth the values of variables in the executing program and, that Holzle2 teaches how invocation counters, which are variables inserted by the dynamic optimization system, not variables in the original program, are used to drive recompilation. Although both are variables, they are very different. The variables in Claim 16 are semantically meaningful to the programmer, i.e., they are part of their algorithm. The variables (invocation counters) in Holzle2 are not created by the programmer and thus, have no semantic meaning in the algorithm. Thus, it is not obvious how the teaching of Holzle2 could be used to derive claim 16 in light of Krishnaswamy.

The Examiner disagrees in the FINAL REJECTION and alleges that the variables in Holzle2 are created by the programmer, and quotes Holzle2 as saying *"Each unoptimized methods has its own counter that is incremented in the method prologue"*. The Examiner further goes on to say *"the counter is a variable and it is related to the method/programmer."*

Respectfully, to clarify applicants' argument, Claim 16 discusses variables that are created by the

programmer, not those created by the system that is the subject of both the current application and Holzle2. Holzle2 creates a variable that tracks invocations (calls) to a method (function). This variable was not created by the programmer of the method (function). Further, Holzle2 does not teach how to record the values of any of the values of variables that are in the original program. To further clarify, applicants offer an example method "Example"

```
Example(int x, char c) {  
    Int y = x * 10;  
    If (y > 10)  
        Print c;  
}
```

The above example has 3 programmer-created variables: x, c, and y. Holzle2's technique would create another variable to record how often the method "Example" was invoked (i.e., called). It does not track the values of x, c, and y. Conversely, Claim 16 is directed to how to track the value of these user-defined variables.

As such, the Examiner is respectfully requested to withdraw the rejection of Claim 16.

Respectfully, the same argument holds for the Examiner's rejection of Claim 17 in which the Examiner relies on the combination of Holzle2 and Krishnaswamy as teaching this claim. That is, the Examiner has argued that Holze2 teaches *"When a counter exceeds the limit, the recompilation driver is invoked to decide which method (if any) should be recompiled"*. Thus, as in Claim 16, this information is not about the original program, but instead is about information added by the Holzle2's system. For example, in the example above, Claim 17 is capturing the control flow path of whether the statement "Print c" was executed. It is not obvious to one skilled in the art how the technique of Holzle2 could capture such paths in the



original program. Thus, it is not obvious how the teaching of Holzle2 could be used to derive Claim 17 in light of Krishnaswamy.

As such, the Examiner is respectfully requested to withdraw the rejection of Claim 17.

While consideration of this amendment is not a matter of right, it appears to applicants that the Examiner, in his rejection of all claims in the Final Rejection dated April 4, 2004, has either not addressed all of the applicants prior remarks set forth in their previous response of January 28, 2004, or, has mischaracterized salient points. Applicants thus respectfully request entry of these amendments and remarks for clarification purposes and, at least, to place the claims in better form for appeal.

This application is now believed to be in condition for allowance, and a Notice of Allowance is respectfully requested. If the Examiner believes a telephone conference might expedite prosecution of this case, it is respectfully requested that the Examiner call applicant's attorney at (516) 742-4343.

Respectfully submitted,



Steve Fischman  
Registration No.: 34,594

Scully, Scott, Murphy & Presser  
400 Garden City Plaza  
Garden City, New York 11530  
(516) 742-4343  
SF:gc